REMARKS

Entry of this Amendment is proper under 37 CFR §1.116 since no new issues or claims are presented.

The Examiner objects to the drawings filed on December 30, 2003, because, as best understood, the replacement drawings failed to include the "Replacement Sheet" label. New replacement drawings will be filed shortly.

Claims 1-20 are all the claims presently pending in the application.

It is noted that Applicants specifically state that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

Claims 4, 5, 9, 10, 15, 16, and 18-20 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. Claims 1, 2, (17), and 20 stand rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 5,438,669 to Nakazawa et al., further in view of US Patent 6,115,730 to Dhablania et al. Claims 3-16, 18, and 19 stand rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over Nakazawa/Dhablania, further in view of Dongarra, et al., "A Set of Level 3 Basic Linear Algebra Subprograms."

These rejections are respectfully traversed in the following discussion.

I.   THE CLAIMED INVENTION

The claimed invention is directed to a software method of improving at least one of efficiency and speed in executing a linear algebra subroutines on a computer having a floating point unit (FPU) capable of overlapping loading data and processing of the data. A load instructions are used to preload data in a timely manner into floating point registers

(FRegs) of the FPU from the L1 cache from an execution code controlling operation of the FPU that is performing the linear algebra subroutines in a near optimal way.

In a programming language, this can be done by "unrolling", which preloads data in advance into the FRegs and then executes corresponding FReg instructions of the linear algebra subroutine.

As explained beginning at line 17 of page 3 of the specification, a number of advancements have been made in the computerized processing of linear algebra routines on computers of various architectures.

The claimed invention addresses one of the problems identified by the present inventors given a specific interface with a recent floating point unit (FPUs) design that typically is used for such linear algebra routine processing.

II.   THE REJECTION UNDER 37 CFR §112, SECOND PARAGRAPH

Claims 4, 5, 9, 10, 15, 16, and 18-20 stand rejected as indefinite. As best understood, the Examiner considers the wording in claims 4, 9, 15, and 18 to lack antecedent basis, and Applicants believe the above claim amendments appropriately address the Examiner's concern.

Relative to the rejection for claim 20, although Applicants are not certain of the basis for the rejection, since the discussion beginning at line 19 on page 12 clearly describes the "five-or-so cycle penalty" at the L1 cache/FPU register interface, Applicants have amended claim 20 in an attempt for clarification.

In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw these rejections.

III.   THE PRIOR ART REJECTIONS

The Examiner alleges that Nakazawa, when modified by newly-cited Dhablania, renders obvious the present invention described by claims 1, 2, (17), and 20, and, when further modified by Dongarra, renders obvious claims 3-16, 18, and 19.

Applicants again submit, however, that there are elements of the claimed invention which are neither taught nor suggested by Nakazawa.

As explained in the second sentence (e.g., in column 1 at lines 12-17), Nakazawa addresses a processing method in a computer architecture in which a cache is not so effective. In this architecture, a large number of data registers accessible to main memory are used. To overcome this design deficiency, Nakazawa introduces preloading of data from main memory directly into these data registers.

In contrast, in the present invention, the L1 cache is used for the matrix data transfer between main memory and the FPUs, so that Nakazawa clearly teaches against the approach of the present invention. That is, the present invention clearly addresses the data preloading into the FPUs using the cache and does so using a software mechanism that does not require the hardware registers of the Nakazawa computer architecture.

The latest rejection fails to address this difference discussed above and, instead, simply introduces secondary reference Dhablania.

In summary, the method in Nakazawa provides a hardware solution to get matrix data from memory to the processor and will work only for the 1995 hardware described therein. In contrast, the present invention provides a general software solution for matrix multiplication, possibly in combination with one or more of the six other co-pending applications.

Therefore, Nakazawa teaches clearly that matrix data using his processing unit cannot be efficiently retrieved from memory using typical cache architecture. In contrast, the present

invention provides basic ground rules for preloading in the context of matrix multiplication kernels. This is novel and not obvious from Nakazawa. That is, relative to independent claim 1 (as well as remaining independent claims), Nakazawa would work only on the now-obsolete 1995 hardware, whereas the present invention is much faster and works with standard cache-based machines.

In the latest rejection, the Examiner adds the following rejection component, in an attempt to answer the plain meaning of the claim language that Applicants noted as being unsatisfied by Nakazawa:

"... *Nakazawa does not explicitly disclose the detailed method about overlapping by preloading data. However, Dhablania in the same analogous art of reloadable floating point unit, discloses a software method of improving at least one of efficiency and speed in executing a linear algebra subroutine on a computer having a floating point unit (FPU) and a load/store unit (LSU) capable of overlapping loading data and processing of said data by the FPU, said method comprising:*

For an execution code controlling operation of said linear algebra subroutine execution, overlapping by preloading data into a̶ floating point registers (Fregs) of said FPU, said overlapping causing data to arrive into said Fregs to be timely executed by the FPU operations of said linear algebra subroutine on said FPU (see for example, Fig. 4a, 4b and related text; also see col. 1, section "Summary of the invention", "ability to initiate a next instruction held in a 4-deep instruction queue before a prior instruction has finished"; col. 4, lines 21-26, "The FPU 70 includes a load/store stage with 4-deep load and store queues")[.]

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the method disclosed by Nakazawa and Dhablania to improve the performance of an FPU by providing it with preload registers which enable

initiation of a next instruction held in a[n] instruction queue as suggested by Dhablania (see for example, col. 1, Summary of the invention)[.]"

In response, Applicants respectfully submits that the above-recited addition to the rejection is clearly engaging in improper hindsight, since there is <u>no reasonable rationale</u> in the latest rejection in accordance with any of the seven rationales proposed by the USPTO in the Federal Register Notices dated October 10, 2007, as published in the aftermath of the US Supreme Court holding in *KSR*. Instead, the rejection merely makes a conclusory statement of the purported benefit of modifying primary reference Nakazawa by newly-cited secondary reference Dhablania.

The problem with the evaluation of record is that the Examiner's initial burden is not satisfied until there is a reasonable rationale articulated on the record. As the US Supreme Court stated in the *KSR* holding: "*'[R]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.'*"

Therefore, in the prior art evaluation of the present invention, since primary reference Nakazawa already has an FPU and a computer architecture based on a large number of data registers, the Examiner has the initial burden to revise this underlying computer architecture in Nakazawa with the FPU of newly-cited Dhablania, using one of the seven rationales recently identified by the USPTO in the Federal Register. Without articulating one of these seven rationales, the rejection currently of record fails to establish a *prima facie* rejection under the *KSR* holding.

More significantly, even if the FPU of newly-cited Dhablania were to be introduced into Nakazawa, the problem being addressed by the present invention would not be solved by this FPU, since, as Applicants pointed out in the previous response and have repeated above and, as clearly described in recently-added dependent claim 20, the present invention is about

11

a penalty between the FPU registers and the L1 cache of newer computer architectures not represented by either Nakazawa or Dhablania, since Nakazawa is clearly based on a large number of data registers, not L1 cache, as Applicants have pointed out.

Newly-cited Dhablania is a patent about hardware. It describes processor technology of the year 1997. In contrast, the present invention relates to processor technology that was introduced during 2004 and later. The present invention overcomes the lack of hardware feature that was not present in 1997. Simply put, Dhablania has little, if anything at all, to do with the present invention.

In the present invention, we need to address the case where the elements in L1 are not contiguous. Dhablania does not apply in the cases that the present invention addresses. Referring to the bullet on page 8 of the latest Office Action, the loads of 2004 and beyond that the present invention is using are load multiple or SIMD load k where k > 1. k is small, typically 2 or 4. These loads happen in a single cycle.

In contrast, for newly-cited Dhablania, even in the good case where the data elements are stride 1, the cost of loading is more than one cycle. In the development of the present invention, the inventors had to live with the lack of a hardware feature. They were able to find a software solution for a large class of matrix algebra subroutines.

One of the aspects that the present invention is addressing is the improvement that secondary reference Dongarra called upon manufacturers to provide, but fails itself to provide a solution.

That is, the present invention improves the performance of the routines that Dongarra has proposed to be needed. The improvement of the present invention is beyond what one would get if Nakazawa were combined with newly-cited Dhablania. The present invention discloses a general technique of preloading that is applicable to several of the routines that Dongarra discloses. Applicants note that Dongarra is only asking for innovation to be

12

supplied by computer vendors.

The present invention supplies such an innovation. It is not found in Nakazawa or in Dhablania, or in their combination.

Hence, turning to the clear language of the claims, in Nakazawa there is no teaching or suggestion of: "A <u>software </u>method of improving at least one of efficiency and speed in executing a linear algebra subroutine <u>in a computer having a floating point unit (FPU)</u> <u>capable of overlapping loading data and processing of said data</u>, said method comprising: for an execution code controlling operation of said FPU performing said linear algebra subroutine execution, <u>overlapping by preloading data</u> into a floating point register (FReg) of said FPU, said overlapping causing data to arrive into said FReg to be timely executed by FPU operations of said linear algebra subroutine on said FPU", as required by independent claim 1. The remaining independent claims have similar language.

Therefore, Applicant submits that there are elements of the claimed invention that are not taught or suggested by Nakazawa. Therefore, the Examiner is respectfully requested to withdraw this rejection for claims 1, 2, and 17.

Claims 2-16, 18, and 19 stand rejected as unpatentable over Nakazawa, further in view of Dongarra. However, regardless of the propriety of combining Dongarra with Nakazawa, this secondary reference does not overcome the basic deficiency identified above for Nakazawa.

Moreover, the following comments relate to the non-obviousness of additional dependent claims over the prior art of record.

Relative to claim 4, as discussed on page 10 of the latest Office Action, Applicants again respectfully point out that use of L1 BLAS and what followed, L2 BLAS, do not work efficiently on today's cache-based architecture. The reason is that they are very slow methods. Nakawaza and Dhablania do not disclose the method of Claim 1. Also, to

paraphrase Dongarra, when he asks suppliers to "build a better mouse trap", the present

invention <u>has</u> built that better mouse trap.

Relative to claim 5, Applicants again point out that Nakazawa and, particularly

Dongarra, do not use level L3 routines for factorization *per se*, and their methods there are

very slow compared to those used in the present invention and the six co-pending

applications. It was not obvious to Dongarra to use fast methods for factorization *per se*.

Relative to claim 18, Dongarra does not disclose any detail on how to implement fast

BLAS. Dongarra is actually asking computer manufacturers to implement his methods,

rather than disclosing fast methods to do this.

Finally, relative to secondary reference Dongarra, Applicants again point out that they

are not attempting to patent matrix multiplication *per se*. Level 3 BLAS are an industry

standard for matrix multiplication. Cayley defined matrix multiplication in 1854 for the first

time. Dongarra merely repeats that definition in very slow implementation of matrix

multiplication.

Therefore, Applicants submit that all claims are clearly patentable over Nakazawa,

even if combined by Dongarra and newly-cited Dhablania, and respectfully request the

Examiner to reconsider and withdraw these rejections.

IV.   FORMAL MATTERS AND CONCLUSION

In view of the foregoing, Applicant submits that claims 1-20, all the claims presently

pending in the application, are patentably distinct over the prior art of record and are in

condition for allowance. The Examiner is respectfully requested to pass the above

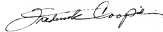application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance,

the Examiner is requested to contact the undersigned at the local telephone number listed

below to discuss any other changes deemed necessary in a <u>telephonic or personal interview</u>.

 The Commissioner is hereby authorized to charge any deficiency in fees or to credit

any overpayment in fees to Assignee's Deposit Account No. 50-0510.

       Respectfully Submitted,

Date: <u>December 17, 2007</u>   _____
           Frederick E. Cooperrider
           Registration No. 36,769

McGinn Intellectual Property Law Group, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254